**Assignment #1**

2015129016 조병준

**- 구체적으로 어떠한 방법과 어떠한 절차를 통해서 학습과 평가 등을 수행했는지**

기본적인 선형회귀모형 학습 및 평가(예측) 절차는 다음과 같다. (세부사항은 3번 질문 답변 참조)

1) 주어진 데이터(‘assignment1\_data.xlsx’) 불러오기 및 numpy array화

2) 종속변수(‘num\_viewers’) 및 독립변수(feature, 제목 및 종속변수 제외한 모든 변수) 구분

3) train\_test\_split 함수를 이용하여 7:3 비율로 학습/평가 데이터 구분

4) LinearRegression() 생성자 함수로 객체(‘model’) 생성 후 학습 데이터에 적합 - 선형회귀모형 학습

5) 평가 데이터에 모형 적용하여 예측력(R^2, 결정계수) 계산

(모형 성능 향상을 위해 추가 적용한 사항은 3번 질문 답변에 기재)

**- 평가데이터에 대한 모형의 성능**

cross-validation과 hyper-parameter tuning을 거쳐 가장 높은 성능을 보인 모형의 평가 데이터에 대한 성능 점수(예측력, 결정계수)는 다음과 같다.

0.7133122527209199 (feature scaling/Min-max scaling 시행, fit\_intercept = False)

(참고. cross-validation 시행한 모형 후보들의 예측력 평균)

1) feature scaling 안 함, fit\_intercept = True, cross-validation(cv) = 5 : 0.7401615873641952

2) feature scaling 안 함, fit\_intercept = False, cross-validation(cv) = 5 : 0.7401615873642048

3) feature scaling/Standardization 시행, fit\_intercept = True, cross-validation(cv) = 5 : 0.7396854047467084

4) feature scaling/Standardization 시행, fit\_intercept = False, cross-validation(cv) = 5 : 0.365497215251456

5) feature scaling/Min-max scaling시행, fit\_intercept = True, cross-validation(cv) = 5 : 0.7401336776950262

**6) feature scaling/Min-max scaling 시행, fit\_intercept = False, cross-validation(cv) = 5 : 0.7401615873641982**

**- 모형의 성능을 높이기 위해 어떠한 노력들을 했는지**

1번 질문의 답변 중 2), 3)번 절차 중 모형 성능 향상을 위해 추가 적용한 사항은 다음과 같다.

a) 독립변수(feature) 선택 - 독립변수 1개~24개(전체)를 차례로 모형에 반영해 예측력을 계산한 결과 데이터셋 내 영화명과 종속변수(‘num\_viewers’)를 제외한 나머지 모든 변수를 독립변수(feature)로 반영하는 것이 가장 높은 점수를 보였음.

b) 학습/평가 데이터 구분 비율 조정 - train\_test\_split() 함수의 인자값을 조정하여 학습/평가데이터 비율을 각각 7:3, 6:4, 8:2로 나눴을 때, 7:3 비율로 구분하였을 때 비교적 높은 예측력을 보였음.

1번 질문의 답변 중 4)번 절차인 ‘선형회귀모형 학습’ 중 모형 성능 향상을 위해 추가 적용한 사항은 다음과 같다.

a) Feature Scaling – Standardization(StandardScaler 함수)과 Min-max scaling(MinMaxScaler 함수) 을 각각 적용하여 독립변수(X\_train, X\_test) 값 변환 후 적용

b) Cross-validation – cross\_val\_score 함수를 이용하여(cv=5, cross-validation folds의 수를 3, 5, 7개로 설정하였을 때, 5개일 경우 비교적 높은 예측력을 보였음) 예측력 계산, standardization scaling 제외한 나머지 모형의 경우 fit\_intercept = False 설정 시 미세하게 더 높은 예측력을 보여주는 것을 확인

c) Hyper-parameter tuning – LinearRegression() 함수로 객체 생성 시 fit\_intercept = False 설정 (y절편 0으로 고정)

**- 본인이 시도한 여러가지 방법 중 어떠한 경우의 성능이 제일 좋았는지, 그리고 해당 방법이 다른 방법들에 비해서 왜 성능이 좋았을 것이라고 생각하는지 등**

cross-validation을 거친 후(cv=5) 각각의 모형 성능을 확인했을 때, 거의 유사하거나 동일한 예측력을 보인 선형회귀모형 후보들이 많았습니다. 가령 Feature scaling을 시행하지 않을 시와 min-max scaling 시행 시 모형의 예측력 평균은 항상 거의 동일했습니다. 또한 fit\_intercept=True 설정 시 min-max scaling 시행과 standardization 시행 경우의 모형 예측력은 거의 동일했습니다. 즉 해당 데이터셋의 경우 feature scaling 시행 여부나 시행 방법이 예측력에 큰 영향을 미치지는 않음을 보여줍니다. 다만 standardization 시행의 경우에 한정하여 fit\_intercept = True 와 False 설정 시 모형의 예측력 차이가 두드러지는데, standardization 시행의 경우에만 fit\_intercept=False 설정 시 예측력이 크게 낮아지는 이유는 다음과 같습니다.

standardization scaling의 경우 독립변수(feature)값의 범위가 [0,1]로 수정되는 min-max scaling와 달리 모든 독립변수의 평균값을 0으로 변환하여 모든 독립변수가 0일 때 종속변수(‘num\_viewers’)가 0이 되는 불가능에 가까운 상황으로 모형이 왜곡되어 모형의 예측력에 부정적인 영향을 미치게 됩니다.

3회 반복 시행해본 결과, feature scaling(Min-max scaling)을 시행하고 fit\_intercept = False로 설정한 선형회귀모형이 항상 가장 높은 적합력(결정계수)을 보이지는 않았으며, 실제 평가 데이터 적용 시 약 0.7133122의 예측력(.score 연산자 참조)을 보였습니다. 그러나 최종적으로 평가데이터에 해당 모형을 적용하기로 판단한 근거는 크게 세 가지가 있습니다.

1. 개별 독립변수마다 다양하게 분포된 값을 균일한 [0,1] 범위 내로 변환해줄 때 coefficients를 통한 각 변수의 영향력 비교가 용이해집니다.
2. feature scaling 미시행 모형과 Min-max scaling 시행 모형의 coefficients를 확인해 볼 때 min-max scaling 시행 모형이 평점(‘netizen\_ratings’)에서 비교적 높은 coefficients를 보였습니다. 즉 관객이 직접 매긴 평점에 대해 민감하게 반응하여 예측하는 모형이 일반적으로 설득력 있을 것이라고 판단했습니다.
3. Min-max scaling 시행 모형의 경우 fit\_intercept = False로 설정하는 것이 fit\_intecept = True로 설정하는 것에 비해 적절해 보입니다. 다시 말해, 모든 독립변수(feature) 값이 0일 때 종속변수(‘num\_viewers’, 음이 아닌 정수) 값이 ‘-828637.35996’이 아닌 0인 모형이 다른 unseen 데이터에 적용하기 용이할 것입니다.